

```

# Loading the important packages

import pickle
import os
import random
import numpy as np
from datetime import datetime

# File to store account data
data_file = "accounts_data.pkl"
transactions_file = "transactions_data.pkl"

# Initialize data
accounts = {}
transactions = []

# Load data from files
if os.path.exists(data_file):
    try:
        with open(data_file, 'rb') as file:
            accounts = pickle.load(file)
    except (pickle.UnpicklingError, EOFError):
        print(f"Error loading account data from {data_file}. Starting with empty data.")

if os.path.exists(transactions_file):
    try:
        with open(transactions_file, 'rb') as file:
            transactions = pickle.load(file)
    except (pickle.UnpicklingError, EOFError):
        print(f"Error loading transaction data from {transactions_file}. Starting with empty data.")

def generate_account_number():
    return str(random.randint(10**10, 10**11-1))

def create_password():
    return str(random.randint(100000, 999999))

def save_data():
    with open(data_file, 'wb') as file:
        pickle.dump(accounts, file)
    with open(transactions_file, 'wb') as file:
        pickle.dump(transactions, file)

def record_transaction(account_no, type_of_accnt, amount,
target_account = None):
    transaction = {

```

```

        "account_no" : account_no,
        "type" : type_of_accnt,
        "amount" : amount,
        "date" : datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
        "target_account" : target_account
    }

    transactions.append(transaction)
    save_data()

# Main Functions
def open_account():
    name = input("Enter account holder's name: ")
    # account_type = input("Enter account type (Savings/Current): ")
    account_type = input("Enter account type (Savings/Current): ").capitalize()
    while account_type not in ["Savings", "Current"]:
        account_type = input("Invalid account type. Please enter 'Savings' or 'Current': ").capitalize()

    initial_balance = float(input("Enter initial balance: "))

    account_no = generate_account_number()
    password = create_password()
    user_id = name.lower().replace(" ", "_") + account_no[-4:]

    accounts[account_no] = {
        "name" : name,
        "type" : account_type,
        "balance" : initial_balance,
        "password" : password,
        "user_id" : user_id
    }

    save_data()
    print(f"Account created succesfully! Account No: {account_no}, User ID: {user_id}, Password: {password}")

def view_account():
    account_no = input("Enter account number: ")
    password = input("Enter account password: ")
    account = accounts.get(account_no)

    if account and account["password"] == password:
        print(f"Account Holder: {account['name']}")
        print(f"Account Type: {account['type']}")
        print(f"Account Balance: {account['balance']}")

```

```

else:
    print("Invalid account number or password.")

def deposit():
    account_no = input("Enter account number: ")
    amount = float(input("Enter amount to deposit: "))

    if account_no in accounts and amount > 0:
        accounts[account_no]["balance"] += amount
        record_transaction(account_no, "Deposit", amount)
        print(f"Deposited {amount} successfully! New Balance:
{accounts[account_no]['balance']}")
    else:
        print("Invalid account number or amount.")

def withdraw():
    account_no = input("Enter account number: ")
    amount = float(input("Enter amount to withdraw: "))

    if account_no in accounts and 0 < amount <= accounts[account_no]
['balance']:
        accounts[account_no]['balance'] -= amount
        record_transaction(account_no, "Withdrawal", amount)
        print(f"Withdraw {amount} successfully! New Balance:
{accounts[account_no]['balance']}")
    else:
        print("Invalid account number or insufficient balance.")

def transfer():
    sender_account = input("Enter sender account number: ")
    recipient_account = input("Enter recipient account number: ")
    amount = float(input("Enter amount to transfer: "))

    if sender_account in accounts and recipient_account in accounts
and accounts[sender_account]['balance'] >= amount and amount > 0:
        accounts[sender_account]['balance'] -= amount
        accounts[recipient_account]['balance'] += amount
        record_transaction(sender_account, "Transfer Out", amount,
recipient_account)
        record_transaction(recipient_account, "Transfer In", amount,
sender_account)
        print(f"Transferred {amount} from {sender_account} to
{recipient_account} successfully!")
    else:
        print("Invalid account numbers or insufficient balance.")

def view_transaction_history():
    account_no = input("Enter account number: ")

```

```

    history = [t for t in transactions if t['account_no'] ==
account_no]

    if history:
        for t in history:
            print(f>Date: {t['date']}, Type: {t['type']}, Amount:
{t['amount']}, Target Account: {t.get('target_account' , 'N/A')}")

    else:
        print("No transactions found for this account.")

def summary_statistics():
    account_no = input("Enter account number: ")
    history = [t for t in transactions if t['account_no'] ==
account_no]

    if history:
        deposits = [t['amount'] for t in history if t['type'] ==
'Deposit']
        withdrawals = [t['amount'] for t in history if t['type'] ==
'Withdrawal']
        print(f"Total Deposits: {sum(deposits)}")
        print(f"Total Withdrawals: {sum(withdrawals)} ")
        print(f"Average Transaction Amount: {np.mean([t['amount'] for
t in history])}")
    else:
        print("No transactions found for this account.")

# Main Menu

def main_menu():
    while True:
        print("\nBank Account Management System")
        print("1. Open a New Account")
        print("2. View Account Details")
        print("3. Deposit Money")
        print("4. Withdraw Money")
        print("5. Transfer Money")
        print("6. View Transaction History")
        print("7. View Summary Statistics")
        print("8. Exit")

        choice = input("Enter your choice: ")
        if choice == '1':
            open_account()
        elif choice == '2':
            view_account()
        elif choice == '3':

```

```

        deposit()
    elif choice == '4':
        withdraw()
    elif choice == '5':
        transfer()
    elif choice == '6':
        view_transaction_history()
    elif choice == '7':
        summary_statistics()
    elif choice == '8':
        print("Thanks! Have a great day.")
        break
    else:
        print("Invalid choice. Please enter a valid option.")

```

Start the program with this command
main_menu()

Bank Account Management System

1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice: 1

Enter account holder's name: Rahul Nagra

Enter account type (Savings/Current): savings

Enter initial balance: 20000

Account created succesfully! Account No: 85031729061, User ID: rahul_nagra9061, Password: 196516

Bank Account Management System

1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice: 1

Enter account holder's name: Ashar Fatmi

Enter account type (Savings/Current): 80000

Invalid account type. Please enter 'Savings' or 'Current': 80000

Invalid account type. Please enter 'Savings' or 'Current': savings

Enter initial balance: 80000
Account created succesfully! Account No: 45967307202, User ID:
ashar_fatmi7202, Password: 499381

Bank Account Management System

1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice: 2

Enter account number: 45967307202

Enter account password: 499381

Account Holder: Ashar Fatmi

Account Type: Savings

Account Balance: 80000.0

Bank Account Management System

1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice: 3

Enter account number: 45967307202

Enter amount to deposit: 5000

Deposited 5000.0 successfully! New Balance: 85000.0

Bank Account Management System

1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice: 4

Enter account number: 45967307202

Enter amount to withdraw: 2000

Withdraw 2000.0 successfully! New Balance: 83000.0

Bank Account Management System

1. Open a New Account

```
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit
Enter your choice: 5
Enter sender account number: 45967307202
Enter recipient account number: 85031729061
Enter amount to transfer: 13000
Transferred 13000.0 from 45967307202 to 85031729061 successfully!
```

```
Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit
Enter your choice: 2
Enter account number: 45967307202
Enter account password: 499381
Invalid account number or password.
```

```
Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit
Enter your choice: 2
Enter account number: 45967307202
Enter account password: 499381
Account Holder: Ashar Fatmi
Account Type: Savings
Account Balance: 70000.0
```

```
Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
```

7. View Summary Statistics
8. Exit
Enter your choice: 6
Enter account number: 45967307202
Date: 2025-01-09 23:46:59, Type: Deposit, Amount: 5000.0, Target
Account: None
Date: 2025-01-09 23:47:23, Type: Withdrawal, Amount: 2000.0, Target
Account: None
Date: 2025-01-09 23:47:43, Type: Transfer Out, Amount: 13000.0, Target
Account: 85031729061

Bank Account Management System

1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice: 7
Enter account number: 45967307202
Total Deposits: 5000.0
Total Withdrawals: 2000.0
Average Transaction Amount: 6666.666666666667

Bank Account Management System

1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice: 8
Thanks! Have a great day.